# User-Centered Planning

## A Discussion on Automated Planning in the Presence of Human Users

**Pascal Bercher, Daniel Höller, Gregor Behnke, Susanne Biundo**

Institute of Artificial Intelligence, Ulm University, Germany

`forename.surname@uni-ulm.de`

### Abstract

We discuss how the *Hybrid Planning* paradigm can be exploited when planning in the presence of human users:

- Intuitive plan generation process (useful for incorporating users into the process, called *Mixed Initiative Planning*).
- Plan execution, which includes:
  - Plan linearization.
  - Presentation of plans.
  - Monitoring of executed steps.
- Plan repair.
- Plan explanation.

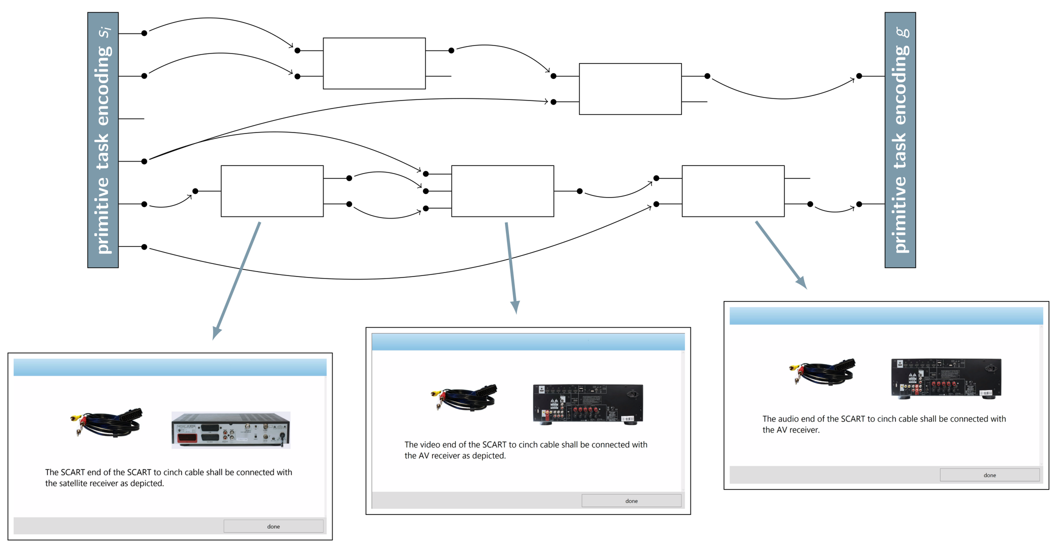
### Example Application: Assembling a Home Entertainment System

To exemplify the discussed user-centered planning capabilities, we use a running example:

**A user wants to assemble his home entertainment system.** The home entertainment system's sub devices must be connected with each other using the correct cables and adapters.

We implemented an assistance system that implements various of the desired user-centered planning capabilities:

- Its instructions are based upon a plan that is automatically generated from a formal description of the hardware.
- The instructions show in detail how to set up the theater.
- The purpose of any presented instruction can be explained.
- The system can cope with execution errors (broken cables).




### Definition (Plan)

A *plan* $P = (PS, \prec, CL)$ is a partially ordered sequence of tasks:

- $PS$ is a finite and possibly empty set of plan steps. Each plan step $l:t$ is a task $t$ with a unique label $l$.
- $\prec \subseteq PS \times PS$ is a strict partial order on $PS$.
- $CL \subseteq PS \times V \times PS$, where $V$ indicates the set of all positive and negative state variables, is a set of causal links between the plan steps. A causal link $l:t \rightarrow_{\varphi} l':t'$ denotes that the precondition $\varphi$ of the plan step $l':t'$ is supported by the plan step $l:t$.

### Definition (Planning Domain and Problem)

A *hybrid planning domain* is a tuple $\mathcal{D} = (T_a, T_p, M)$, where:

- $T_a$, $T_p$ are finite sets of abstract and primitive tasks, respectively. Each (primitive or abstract) task is a pair $(prec, eff)$ consisting of a conjunction of literals over the set of state variables.
- $M$ is a finite set of *(decomposition) methods.* A method $m = (t, P)$ maps an abstract task $t \in T_a$ to a plan $P$.

A *hybrid planning problem* is a tuple $\mathcal{P} = (\mathcal{D}, s_i, P_i, g)$, where:

- $\mathcal{D}$ is the planning domain.
- $s_i$ and $g$ are the initial state and the goal description, respectively.
- $P_i$ is the initial plan. As usual in POCL planning, it contains two special actions that encode $s_i$ and $g$.

### Definition (Solution Plan)

A plan $P$ is a solution iff:

- $P$ is a refinement $P_i$, i.e., $P$ can be obtained from $P_i$ via:
  - Decomposition:
    given a plan $P' = (PS, \prec, CL)$, use method $(t, P'') \in M$ to replace $l:t \in PS$ by $P''$. Causal links and orderings are inherited.
  - Insertion of ordering constraints.
  - Insertion of tasks. *This refinement is optional!*
  - Insertion of causal links.
- Every linearization of $P$ is executable in $s_i$ and satisfies $g$.

*In the absence of causal links, the respective problem classes are called HTN planning or TIHTN planning (HTN planning with task insertion), depending on whether the insertion of tasks is allowed.*

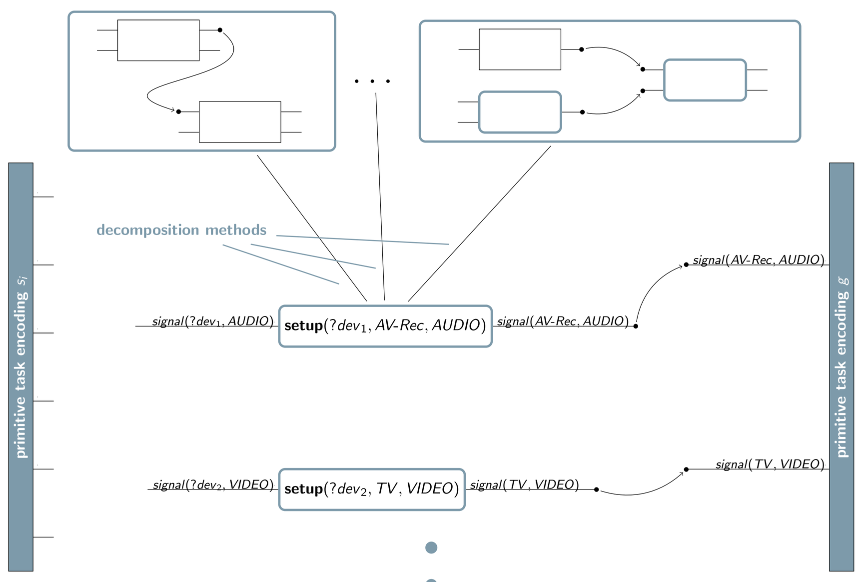
### Intuitive Plan Generation Process

How are solutions generated and why and how is this process beneficial for planning with or for humans?

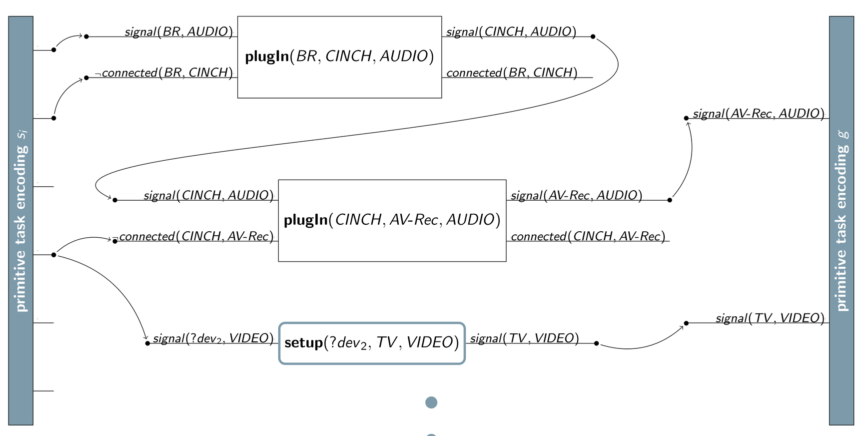
By step-wise refining the initial plan into a solution plan, a user can be smoothly integrated into the plan generation process.

- Hierarchical Refinement:
  - Abstract tasks are step-wise refined into more primitive courses of action – similar to human problem solving.
  - Abstract tasks show preconditions and effects, which gives them a clear semantics and allows to generate plans "on an abstract level".
- Goal-Directed Refinement:
  - Missing steps are inserted by analyzing causal dependencies in a goal-directed way.
  - Causal dependencies are explicitly represented via causal links. That way "unfinished" parts of the plan can be identified.

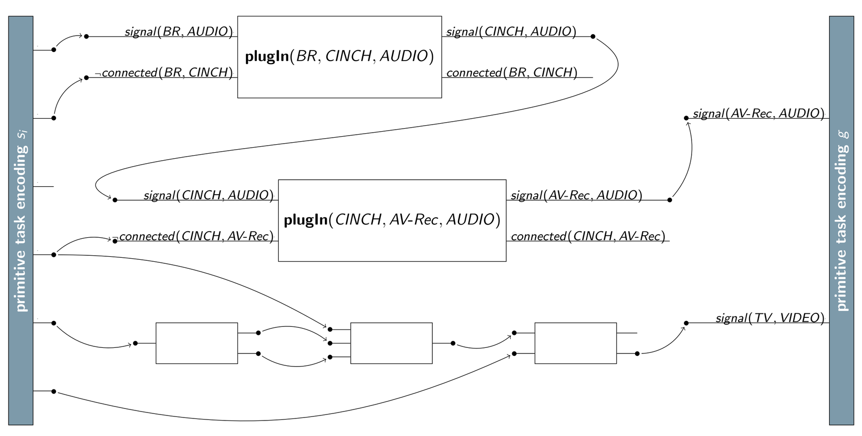
**Initial plan $\mathbf{P_i}$**
(the graphic also shows some decomposition methods)




**Intermediate plan**




**Solution plan**




### Plan Execution

What does it mean to execute a plan?

- Actions may be executed automatically by a system (like a mobile phone, a computer, or an intelligent environment).
- Actions may be executed by a user. For that purpose, they need to be presented, first.

In any case, the partial order needs to be linearized. Then, the actions can be communicated to the user.

### User-friendly Plan Linearization

The solution criteria of hybrid planning ensure that *any* linearization of the solution plan is an executable action sequence. However, some of them might me more plausible for humans than others. This raises the question which of the linearizations to pick.

Consider the following two *valid* linearizations of our example:

- 1: **plugIn**(*BR,CINCH,AUDIO*),
  2: **plugIn**(*CINCH,AV-Rec,AUDIO*),
  3: ..., 4: ..., 5: ...
- 1: ...,
  2: **plugIn**(*BR,CINCH,AUDIO*),
  3: ...,
  4: **plugIn**(*CINCH,AV-Rec,AUDIO*),
  5: ...

Clearly, the first linearization is more plausible for a human user, since it finishes connecting both ends of the CINCH cable before moving on to another cable or device.

We have developed three domain-independent strategies to find plausible, user-friendly linearizations – they are based on the features of the hybrid planning paradigm:

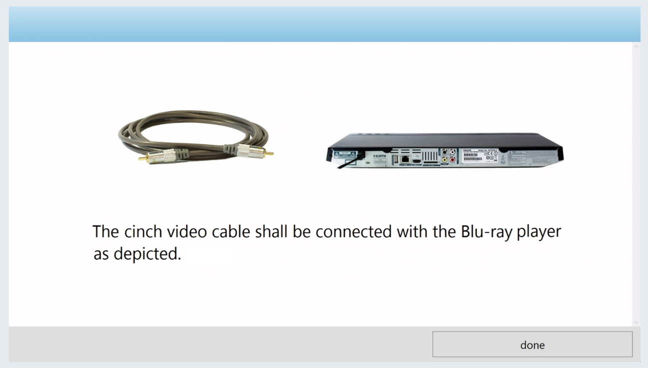
- On task parameter similarity.
- On the causal structure.
- On the task hierarchy.

### Plan Presentation

After an execution sequence has been selected, the actions have to be communicated to the user (assuming they are not automatically executed by a system).

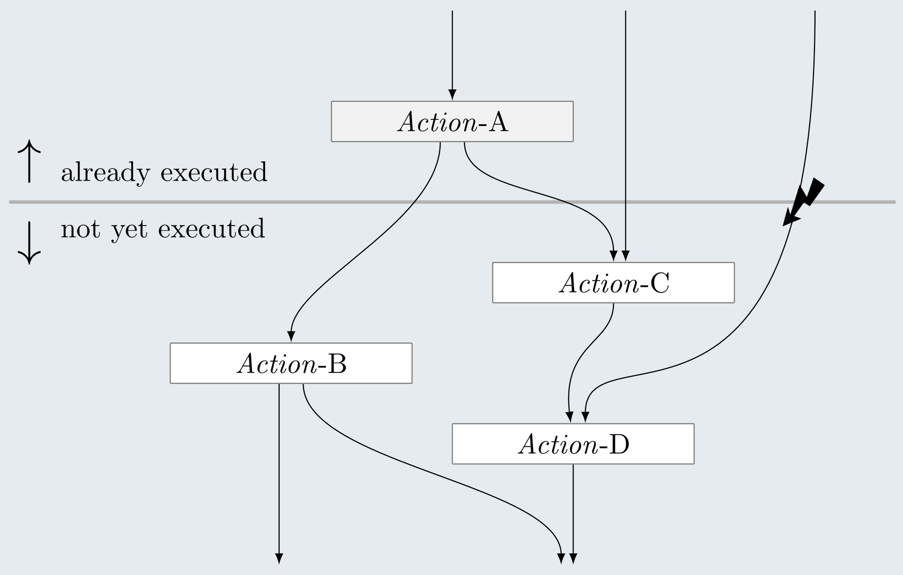
Consider the action **plugIn**(*BR,CINCH,AUDIO*).

Given that the knowledge base stores appropriate pictures and videos for the objects used by the action, detailed interfaces can be generated automatically. Using templates, the instructions can also be presented using natural language.




### Plan Repair

The plan execution component monitors the execution state of the plan: in case the current state deviates from the anticipated one, plan repair is initiated.




(note: the plan is rotated by 90 degrees: it is ordered from top to bottom)

Properties of plan repair:

- The repaired plan is still a solution to the original problem.
- All executed steps are contained in any new solution (marked as executed), so one does not have to start over.

Issues of plan repair:

- When to repair? Also when "short cuts" are possible?
- How to repair? Plan stability/similarity vs. optimization criteria.

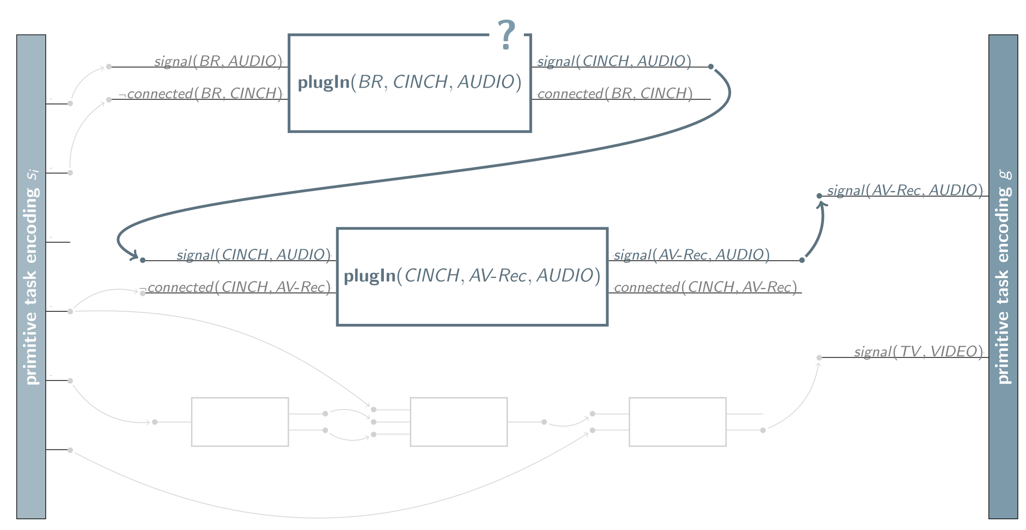
### Plan Explanation

To obtain full transparency of a system, it needs be able to justify its behavior/decisions. To that purpose we can explain certain properties of plans, e.g.:

- Why has action $A$ to be performed before $B$?
- Why does action $C$ manipulate object $o$?
- Why is action $D$ part of the plan, anyway?

These questions can be answered in natural language. This is done based on a proof in an axiomatic system. That system formalizes:

- The decomposition hierarchy
- The solution plan's causal structure (causal links)




### Plan Explanation (Example)

In the plan depicted above, the user wants to know why he should perform action **plugIn**(*BR,CINCH,AUDIO*).

The analysis of the causal structure (highlighted in the depicted plan) corresponds to a sequence of single proof steps:

- cr( **plugIn**(*BR,CINCH,AUDIO*) , **plugIn**(*CINCH,AV-Rec,AUDIO*) )
  $\wedge$ n( **plugIn**(*CINCH,AV-Rec,AUDIO*) )
  $\Rightarrow$ n( **plugIn**(*BR,CINCH,AUDIO*) )
- cr( **plugIn**(*CINCH,AV-Rec,AUDIO*) , **goal-task-for-g** )
  $\wedge$ n( **goal-task-for-g** )
  $\Rightarrow$ n( **plugIn**(*CINCH,AV-Rec,AUDIO*) )

These proof steps can be translated into natural language:
"You have to connect the Blu-ray player with the CINCH cable to be able to connect that cable with the AV receiver, so the audio signal is transmitted from the Blu-ray player to the AV receiver.